Regressão logística - Exercícios

Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas - 2020/01
Prof. Eric Aislan Antonelo

1 Exercícios

(a) Adapte o código da regressão linear para implementar um modelo de regressão logística para classificação binária em 1 dimensão. O esqueleto do código pode ser visto abaixo. Implemente as funções apresentadas. Escolha um valor inicial para θ. Defina a taxa de aprendizagem.

Realize o treinamento do modelo com o algoritmo do descenso do gradiente, considerando o modelo de regressão logística de uma variável:

$h_\theta(x) = g(\theta_0 + \theta_1 x)$

A tarefa de classificação consiste em encontrar o valor dos parâmetros com a melhor fronteira de decisão que separa a classe 0 da classe 1, maximizando a acurácia (ou taxa de classificação) da hipótese treinada. Lembre-se que a saída final do classificador é binária.

```
import numpy as np
from matplotlib.pyplot import subplot, plot, show, clf, vlines
import matplotlib.pyplot as plt

# conjunto de dados {(x,y)}

mean0, std0 = -0.4, 0.5
mean1, std1 = 0.9, 0.3
m = 200

x1s = np.random.randn(m//2) * std1 + mean1
x0s = np.random.randn(m//2) * std0 + mean0
xs = np.hstack((x1s, x0s))

ys = np.hstack(( np.ones(m//2), np.zeros(m//2)))

plot(xs[:m//2],ys[:m//2],'.')
plot(xs[m//2:],ys[m//2:],'.')
show()

def sigmoid(z):
    pass

''' hipotese
    sigmoid(theta[0] + theta[1] * x)
'''
def h(x, theta):
    pass

''' funcao de custo para um exemplo; entropia cruzada
'''
def cost(h, y):
```

```
    pass

''' funcao de custo total
'''
def J(theta, xs, ys):
    pass

''' derivada parcial com respeito a theta[i]
'''
def gradient(i, theta, xs, ys):
    pass

def plot_fronteira(theta):
    # use vlines() para plotar uma reta vertical
    pass

''' plota em subplots: - os dados, com a fronteira de decisao
    - e os dados classificados
'''
def print_modelo(theta, xs, ys):
    pass

def accuracy(ys, predictions):
    num = sum(ys == predictions)
    return num/len(ys)


alpha = None # completar
epochs = 600
theta = None # completar

for k in range(epochs): # 10000

    # apply gradient decent

    # show classication performance
    print('Acuracia: ', accuracy(ys, predictions) )
```

(b) O que ocorre com a acurácia ao longo do processo de treinamento? Consegue obter 100% de acurácia? Por quê?

(c) Plote a função de custo ao longo das iterações (épocas) por 2000 épocas de treinamento. Em outros dois plots, desenhe a acurácia e a localização da fronteira (valor que divide os pontos x), como no gráfico abaixo (Fig. 1):

(d) Experimente com valores altos e baixos para taxa de aprendizagem. O que acontece? Notou alguma diferença para valores altos com relação a regressão linear?

(e) (opcional) Plote a função de custo $J(\theta)$ em função de θ.

(f) (opcional) Modifique o código para implementar regressão logística para 2 variáveis de entrada (x_1, x_2).

(g) (opcional) Plote a fronteira de decisão (reta) junto com os dados em duas dimensões, para o caso de 2 variáveis.

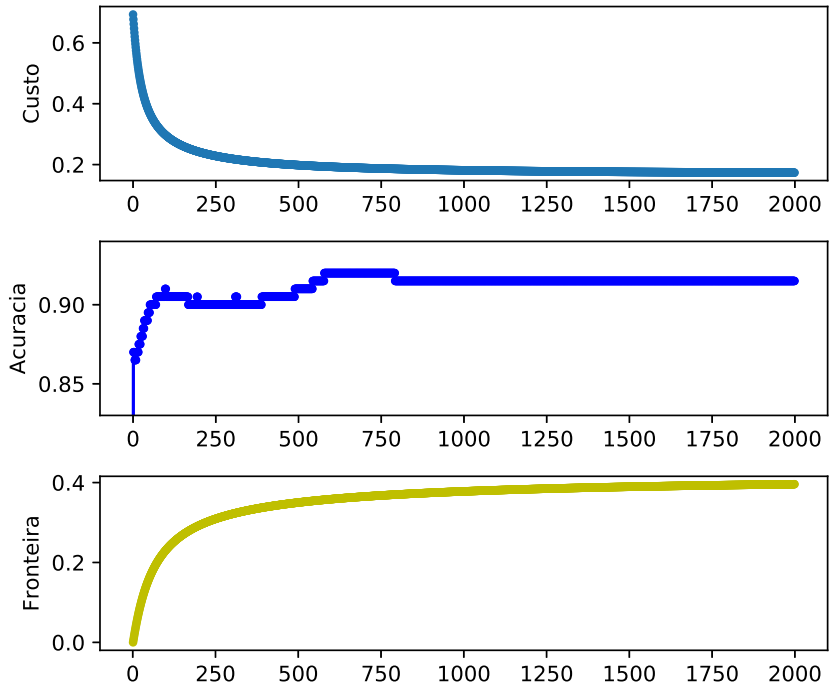

Figure 1: Custo, acurácia, e localização da fronteira ao longo do tempo.